

Network awareness in P2P-TV applications

Original

Network awareness in P2P-TV applications / Traverso, Stefano; Leonardi, Emilio; Mellia, Marco; Meo, Michela. - 5733:(2009), pp. 41-50. (Intervento presentato al convegno Eunice 2009 tenutosi a Barcelona nel 7-9 settembre 2009) [10.1007/978-3-642-03700-9_5].

Availability:

This version is available at: 11583/2358631 since:

Publisher:

Springer-Verlag

Published

DOI:10.1007/978-3-642-03700-9_5

Terms of use:

openAccess

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

Network awareness in P2P-TV applications [★]

S. Traverso, E. Leonardi, M. Mellia, and M. Meo

Politecnico di Torino lastname@tlc.polito.it

Abstract. The increasing popularity of applications for video-streaming based on P2P paradigm (P2P-TV) is raising the interest of both broadcasters and network operators. The former see a promising technology to reduce the cost of streaming content over the Internet, while offering a world-wide service. The latter instead fear that the traffic offered by these applications can grow without control, affecting other services, and possibly causing network congestion and collapse. The “Network-Aware P2P-TV Application over Wise Networks” FP7 project aims at studying and developing a novel P2P-TV application offering the chance to broadcast high definition video to broadcasters and to carefully manage the traffic offered by peers to the network, therefore avoiding worries to Internet providers about network overload. In such context, we design a simulator to evaluate performance of different P2P-TV solutions, to compare them both considering end-users’ and network providers’ perspectives, such as quality of service perceived by subscribers and link utilization. In this paper, we provide some results that show how effective can be a network aware P2P-TV system.

1 Introduction

Last years witnessed the emergency of many P2P video-streaming services (P2P-TV) on the Internet, like SopCast [1] or PPLive [2] to possibly name the most popular ones. Recently, a new and promising generation of high-definition commercial video P2P applications, such as Babelgum [3], Zattoo [4] or TVUnetworks [5], are at an advanced stage of development and testing. These systems are targeted to offer high bandwidth video streams (1 to 5 Mbit/s) to a large population of users (up to millions). These applications are seen as a good opportunity by broadcasters to both reduce the cost of providing a streaming service to the Internet, and to reach a world-wide population of users. However, the same motivations constitute a worry for network carriers since the traffic P2P-TV application can generate may potentially grow without control, causing a degradation of quality of service perceived by users or even a collapse of network functionalities. This is the main research topic of the FP7 project “Network-Aware P2P-TV Application over Wise Networks”, which aims at studying a novel P2P-TV application that offers high quality service to the end-users, while guaranteeing network providers to optimize network resource usage. Therefore, to study, characterize and optimize P2P-TV service, we developed a simulator that allows us to understand the impact of these new services on network performance: the main objective of this paper is to propose

[★] This work was funded by the European Commission under the 7th Framework Programme Strep Project “NAPA-WINE” (Network Aware Peer-to-Peer Application over Wise Network)

a simple evaluation of P2P-TV systems, focusing on their impact on the transport network.

A preliminary distinction must be done: when using a general P2P application, two different *topologies* have to be considered: i) a *logical topology* at application level, made up by peers, which is created and updated whenever a peer joins or leaves the network; and ii) a *physical network* composed by hosts, nodes and links, which is typically given and rarely changes. In other words, each peer can be seen as a P2P application running on a PC; it establishes logical connections with other peers running on some other PCs, which are then used to exchange the content, e.g., the video in case of P2P-TV applications. The graph which is created by peers is called *Overlay topology*. Considering, instead, the physical network, it is made up by real devices like routers and links, and the graph they form is called *Underlay topology*.

Typically, the Overlay and Underlay topologies are independently set up, managed and optimized. For example, in BitTorrent, a popular P2P file sharing application, a peer selects to which peers upload the content based on the amount of data received in the past, playing the famous tit-for-tat algorithm [6]. However, if the selected peer is actually physically close or far away is not taken into any consideration. The tit-for-tat algorithm clearly optimizes application layer performance, e.g., maximizing download throughput, but it possibly causes a large waste of resources at the Underlay topology, e.g., downloading data from a high-speed peer far away, thus forcing the network to transport traffic over several links.

Similarly, P2P-TV applications transmit streaming videos on the Overlay topology ignoring information about the Underlay network, so that the Overlay topology is built “randomly”, without any awareness about location or available bandwidth of hosts which run the application. In this paper we quantify the lack of efficiency due to this issue and demonstrate that the adoption of smart strategies in Overlay topology creation process can improve performance of both the application and the network. We focus our attention on P2P-TV systems, in which the Overlay topology is a generic mesh, and a swarm-like delivery of the video is adopted: a source node splits the video stream into small *chunks*, which are then transmitted to and by peers, exchanging chunks according to some scheduling scheme.

We first define a simple, yet accurate model which describes both the P2P-TV application and the transport network. Then, we compare performance of different algorithms with increasing knowledge about topological information of the network layer exploited to set up the P2P Overlay graph. Results show that it is possible to reduce the traffic the network has to carry by more than two order of magnitudes, while at the same time improving quality of service users perceive.

2 Scenario Description

In the following, we briefly describe the reference model that has been implemented in the simulator. More details are available in [7]. First we describe the model of the Underlay topology, which is assumed to be given. Then the Overlay topology and chunk scheduling algorithms are described.

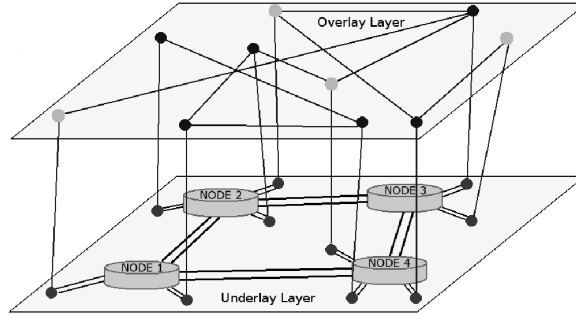


Fig. 1. Overlay and Underlay topologies (RC strategy applied).

2.1 The Underlay topology

As already mentioned, we have to consider two different topologies: the first one is built at application level, and it is composed by peers which set up connection among them to transmit chunks. Peers are actually hosts, connected to the Internet, whose topology constitutes the Underlay topology. Here, *nodes* (or routers) are connected by means of bidirectional physical links, with a given capacity. Overlay connections correspond to “paths” at the Underlay topology, so packets forming a chunk are routed through the Underlay topology from the source host up to the destination host, following for example the shortest path.

Fig. 1 shows an example of a Underlay topology (at the bottom) in which four routers are connected forming a ring topology among them. Hosts (or users) are then connected to a router. Hosts are running the P2P-TV application, and therefore they are peers, which are connected forming the Overlay topology (at the top).

In the simulator, the chunk transmission by a host is modeled according to a fluid traffic model, so that no packet is actually routed. Instead, on all links along the path, a given amount of bandwidth will be allocated for the duration of the data transmission. Hosts transmit one chunk at a time; given the chunk size and the upload capacity of a host, an amount of bandwidth comparable to the chunk transmission speed is allocated on all links along the path to the destination for the whole chunk duration. This model allows to get therefore the actual link load at any time, given the set of chunks that are actually transmitted by peers. At the same time, it avoids the burden to simulate packet level details, therefore allowing to consider larger networks and higher number of peers. In [7], we were able to run simulations with hundreds of nodes, several thousands of peers transmitting video streams of thousands of chunks.

In this paper, we consider as Underlay topology a simplified version of the actual Telecom Italia backbone network [8]: two main routers are placed in Milano and Roma, and they are connected by 40 Gb/s links. Those two routers and links form the “backbone” of the topology. Nodes in cities in the north part of Italy are then connected to node in Milano, while nodes in the south part of Italy are instead connected to the backbone node in Roma. 10 Gb/s links are used. Nodes in each city are called “access routers”, and there are twenty of them in the considered scenario. Hosts are then connected to access routers using either a xDSL link, or using higher capacity connections,

like FTTH access. For the sake of simplicity, we assume that two classes of hosts are present: i) *residential* users, with xDSL connection of about 0.5 Mb/s upload capacity, and ii) *business* users, with high access bandwidth of about 5 Mb/s. We refer to residential and business peers as Low-Speed (LS) and High-Speed (HS) peers respectively in the following. In both cases, we assume the download capacity is large enough so that the bottleneck is the uplink capacity of hosts. There are 500 hosts globally, each of them randomly connected to a selected access router, so peers are uniformly distributed over the Underlay topology.

2.2 The Overlay topology

As introduced in Sec. 1 the topology which is created at application level is a graph composed by peers and logical connections. Given a peer, there is a subset of all peers which are then connected to it, called “neighborhood”.

Each peer belonging to the Overlay topology is permanently associated to a host belonging to the Underlay topology - Fig. 1. In our model, we assume each peer establishes on average k logical connections with other peers to build the graph which represents the Overlay topology; k is called “degree” of peer. In this paper, we set an average degree $k = 8$, independently on the Overlay topology generation strategy we adopted (see below). HS-peers are granted an additional number of neighbors k' . The intuition behind this is to offer peers with more upload capacity the chance to serve more neighbors. For the sake of simplicity, *churning* phenomenon is not considered in this paper, so the number of peers involved in simulation is fixed and never changes: no peers are expected to join or leave the system during the simulation. In this paper, we compare three different algorithms to setup the Overlay topology, which entail an increasing level of “awareness” by the peers:

- **Random Choice strategy - RC:** In this case, each peer selects on average k neighbors at random, therefore completely ignoring the actual location of hosts in the Underlay topology. This models the today typical approach in which peers ignore any information about the network layer, and Overlay topology results uncorrelated to the actual Underlay topology.
- **Location and Bandwidth Awareness strategy - LBA** In this case peers have some knowledge about the physiognomy of the Underlay topology, so that peers are connected to the topologically closest peers in order to avoid congestion on network backbone links and to reduce the average delivery time of chunks. In particular, in this model two peers are topologically close if the corresponding hosts are connected to the same access node. Each access node represents an “isle” of peers, in which peers are connected among them using a Random Choice policy. To interconnect the island among them, HS-peers *randomly* connect between them using additional k' connections, forming therefore a hierarchical topology. In other words, the Overlay topology exploits information about the peer location to form groups of close peers, and information of peer upload capacity to interconnect different groups of peers.
- **Neighbored Location and Bandwidth Awareness strategy - NLBA** In this case, the HS-peers perform a smarter choice when selecting other k' HS-peers, so that

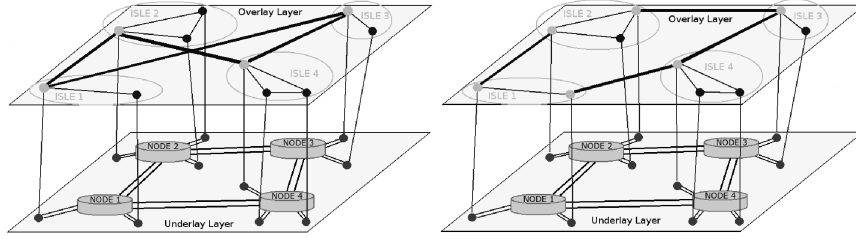


Fig. 2. Examples of LBA and NLBA Overlay topologies on the left and right figures.

physically closer peers are selected. The resulting Overlay topology therefore entails a greater knowledge about the Underlay topology, so that the communication between island is optimized as well.

Fig.2 shows two examples of Overlay topologies in which the LBA and NLBA policies are adopted (left and right figures respectively). HS-peers high are represented with light-coloured dots, while dark dots represent LS-peers. The NLBA topology better reflects the Underlay topology, so that HS-peers are interconnected among them mimicking the same ring topology which is present in the Underlay topology.

2.3 Chunk Scheduling algorithm

Once the Overlay topology has been created, peers can then start exchanging video chunks. A *source* node encodes the video stream, chops it into chunks, which are regularly injected in the Overlay by transmitting them to source neighbors; then, peers that received a new chunk can upload it to those neighbors that have not yet received it, according to a “scheduling” algorithm. We assume a push mechanism, in which the transmitter selects both which chunk to transmit and to which peer. Several scheduling algorithms have been proposed in the literature (see for example [9–12]), among which we selected the simplest one since our focus is on the Overlay topology optimization rather than chunk scheduling algorithm. Therefore, we adopt a simple *random chunk/random peer* scheduling scheme: each peer selects at random one chunk among those it has received and still stores in the trading window ([12] employs a sliding window mechanism to optimize chunk transmission); then it selects at random one of its neighbors among those that have not yet received the selected chunk. The peer then transmits the chunk to the selected peer. When the transmission ends, a new chunk and a new peer are selected. In P2PTV terminology this policy is called RND-RND scheduling policy [12]. Note that the transmission time depends on the chunk size and on the transmitter upload capacity.

All the above algorithms have been implemented in the simulator. Details about the simulator design and performance are available from [7, 14].

2.4 Simulation scenario

Here we summarize all the parameters that define the simulation scenario considered in this paper. The video-rate is assumed to be equal to 0.4 Mb/s, and the chunk size

is fixed to 2 Mbit. Therefore, the source node (selected at random among one of the HS-peer) generates a new chunk every 5 seconds. Each peer has an average number of neighbors equal to 8 neighbors ($k = 8$) for all Overlay construction strategies. For all tests, simulations involve 500 peers, and the simulation lasts 500 chunks, i.e., 2500s of video.

The capacity of links in the physical topology is very large (10Gb/s or 40Gb/s), so that the bottleneck link is the upload capacity of each peer. This is a typical scenario in which users are connected to the Internet with xDSL links, in which the download capacity is typically higher than the upload capacity. Upload access bandwidth is 0.5 Mbit/s for LB-peers and 5 Mbit/s for HS-peers. Download bandwidth was equal to 5 Mbit/s for every peer. The chunk scheduling algorithm is RND-RND, with a trading window size equal to 5 chunks. We run simulation then to compare the three different Overlay topology construction policies: RC, LBA and NLBA strategies. As parameter, we vary the additional degree of HS-peers k' , and the number of HS-peers that are present in the Underlay topology.

As performance indexes, we select the load on actual links of the Underlay topology, i.e., the load due to chunk transmission on the physical link. We report also the average number of hops traversed by each chunk, i.e., the corresponding Underlay path length of each Overlay link. Both these two metrics are important to assess the impact on the Underlay topology of the P2P-TV traffic, so to appreciate how increasing degree of network awareness can reduce the actual traffic the Underlay network has to carry. All results presented in this paper are averaged over 10 independent runs.

To observe the impact of the different topologies on the quality of service perceived by users, we evaluate the *chunk delivery delay*, i.e., the delay from when the source emits the chunk to when a peer receives it. Indeed, assuming that there is enough capacity to deliver the video stream to all peers, the chunk delivery delay is typically considered the main performance index to be optimized in live P2P-TV systems: minimizing it means minimizing the delay since when the information is produced at the source and when it can be played at the peers, guaranteeing therefore better “live” experience; furthermore, the chunk delivery delay is also related to the startup time, i.e., the time required to a peer that “tunes” to a channel to start receiving the video data [12].

3 Results

3.1 Network-centric scenario

Fig.3 reports the network link load for the different Overlay topology strategies, using a log scale. Plot on the left refers to the Underlay topology backbone links, i.e., the links between Milano and Roma, while the plot on the right refers to the access links, i.e., the links between access and backbone nodes. The additional degree of HS-peers k' is used as parameter. It can be observed that both LBA and NLBA reduce the bandwidth occupation on network links of a factor of 10. In particular, forming the island between peers that are connected to the same access nodes (LBA algorithms) allows to reduce the amount of traffic in backbone links from 45Mb/s (corresponding to about 115 video streams) to only 5Mb/s (or about 12 video streams). This is due to the fact that most of

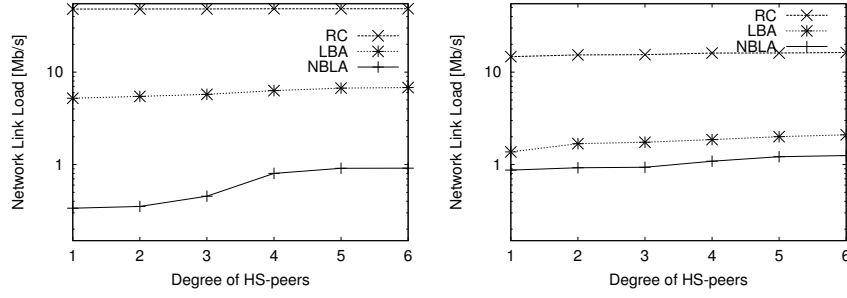


Fig. 3. Network links load adopting different strategies for varying degree k' of HS-peers. Backbone links (links between Milano and Roma spots) on the left, access links (all node-node links which are not backbone) on the right.

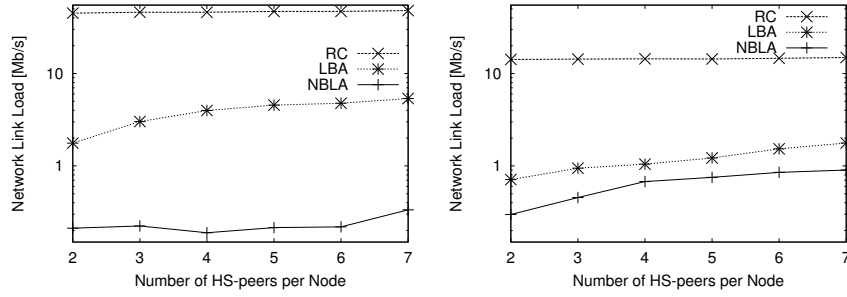


Fig. 4. Network links load adopting different strategies for varying number of HS-peers per node. Backbone links on the left, access links on the right.

the chunks are transmitted between peers that are connected to the same access node, therefore avoiding crossing backbone links. Enforcing a smarter interconnection between HS-peers (NLBA algorithm) further reduces the load on backbone links, so that about 0.4Mb/s is on average transmitted, corresponding to the minimum value of a single copy of the video stream. Notice that the degree of links between HS-peers has little effects on the link load. Indeed, it is sufficient to use one additional link to already optimize the Overlay performance. Moreover, increasing k' actually reduces the benefit of the optimized topology, since two HS-peers that are far apart in the Underlay topology can possibly exchange chunks due the random peer selection policy implemented by the chunk scheduling algorithm.

Similar considerations hold when looking at the load carried on access links. In this case it is possible to note that the improvement of the NLBA and the LBA strategies are more similar. Indeed, the choice of the k' additional neighbors has been designed to reduce the traffic on the backbone links only.

Fig.4 reports the network link load versus the number of HS-peers per actual Underlay node. Also in this case it can be seen that the presence of at least one HS-peer is sufficient to optimize the Overlay topology so that traffic on the Underlay network is reduced (in this case we selected $k' = 2$). Similarly as in the previous case, increasing

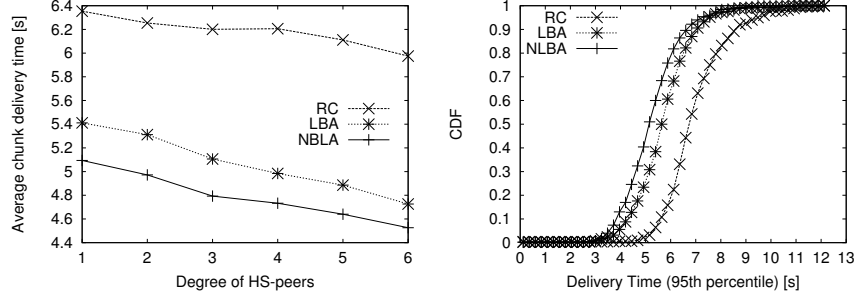


Fig. 5. Average chunk delivery time for varying degree k' of HS-peers on the left. Cumulative distribution function of the 95 percentile of the chunk delivery time for different Overlay topology strategies on the right.

the number of HS-peers per nodes increases the network link load as well. This is due again to the chunk scheduling policy that allows a random selection of peers. In case a larger number of HS-peers is present, a larger number of links among them in the Overlay topology is present. During chunk scheduling, then, there is an increased probability that those links are selected to transmit a chunk between two HS-peers. Finally, in case the LBA strategy is considered, HS-peer to HS-peer connections are generated at random, possibly between two far apart nodes, so that more physical links are crossed.

To confirm this intuition, and to appreciate the better usage of Underlay resources, the number of hops traversed by each successful chunk transmission, $E[H]$, is reported. Practically, this parameter shows how many “local” transmissions a strategy can do. Simulations with configuration described above shows that the mean number of hops per each chunk transmission adopting a RC strategy is

$$E[H]_{RC} = 3.11 \quad (1)$$

whereas much better results are reached by smart algorithms:

$$E[H]_{LBA} = 1.33, \quad (2)$$

$$E[H]_{NLBA} = 1.14 \quad (3)$$

These results confirm that most of chunks do not traverse more than one hop when location awareness is enabled. NLBA strategy gives best results because logical links among HS-peers, that connect together different islands, are mapped at most on only two hop long path, corresponding to a (host)-(node), a (node)-(node), and finally a (node)-(host) on the Underlay topology; moreover, all other links between two LS-peers or between a LS-peer and a HS-peer are mapped on (host)-(node) and (node)-(host) links, so that only one router is crossed.

3.2 User-centric performance

Fig. 5 reports on the left the average chunk delivery time for varying degree of HS-peers and for different strategies. The number of peers in the simulation is 1000, with 6 HS-peers for each node. Few considerations hold: first, both LBA and NLBA offer better

performance than the RC policy. This is due to the presence of connections between HS-peers that form a fast backbone among HS-peers. This allows to quickly distribute the chunks on the Overlay topology [12]. Indeed, in the RC case, the additional k' connections HS-peers have are randomly spread among all peers. On the contrary, the Bandwidth awareness of LBA and NLBA strategies forces HS-peers to preferentially connect to other HS-peers. In addition, Fig. 5 shows that increasing k' improves the probability of chunks to be transmitted between HS-peers, reducing chunk delivery time.

To give more insight, Fig. 5 reports on the right the cumulative distribution function (among peers) of the 95th percentile of the chunk delivery time. For each peer we compute 95th percentile of the chunk delivery time (computed among all chunks), and plot its distribution among peers. We select this index as another good parameter to quantify the quality of service perceived by each user. Intuitively, it reports the delay with which peers receive the chunk with probability 0.95. The results refer to a scenario in which $k' = 2$ and the number of HS-peers per node is set to 6 as above. As we can see, again NLBA and LBA strategies obtain better performance respect to the RC topology, even if improvements are limited. This is essentially due to the fact that LBA and NLBA strategies prefer to interconnect HS-peers among themselves, forming a sort of high speed interconnection that allows to quickly spread the chunks among the islands of peers. On the contrary, the RC strategy does not explicitly exploit the peer bandwidth when interconnecting peers. Moreover, the Random peer and chunk selection ignores any peer properties, so that the resulting chunk spreading is not optimal. In more details, after six seconds both NLBA and LBA strategies guarantee that 80% of peers has already received the chunks in 95% of time. Adopting the RC strategy, after 6 seconds, only 20% of peers are assured to receive the chunk with probability 0.95.

In summary, the network aware Overlay topologies can also improve the P2P-TV performance, while at the same time greatly reducing the offered traffic the Underlay network has to carry.

4 Conclusions

In this paper we investigated the impact of P2P-TV traffic over the physical topology of a network. Since P2P-TV applications are increasingly becoming popular among users, networks operators are worried that the traffic these applications can inject in the network may cause potential congestion if not carefully controlled. We therefore developed a simulator to efficiently study the amount of traffic links of the network carry due to peers watching a P2P-TV stream. We compared then different Overlay setup strategies with increasing level of network-awareness, i.e., in which information about the physical placement of peers and their capacity is made available to the peers during the overlay topology setup.

Results, even if preliminary, show that it is possible to reduce the amount of traffic the network has to carry by up to two orders of magnitude, while, at the same time improving the quality of service perceived by users. This encourage further investigations to increase the level of network awareness of P2P-TV applications.

References

1. SopCast, <http://www.sopcast.com>.
2. PPLive, <http://www.pplive.com>.
3. Babelgum, <http://www.babelgum.com>.
4. Zattoo, <http://www.zattoo.com>.
5. TVUnetworks, <http://www.tvunetworks.com>.
6. B. Cohen "Incentives Build Robustness in BitTorrent", May 2003.
7. S. Traverso, Master Thesis: Design and Implementation of an Integrated Simulator for P2P Video-streaming Application Performance Evaluation, Politecnico di Torino, 2008.
8. A. M. Langellotti S. Mastropietro F. T. Moretti A. Soldati, Il Backbone IP di Telecom Italia Wireline, Notiziario Tecnico Telecom Italia Anno 13 n. 2 - Roma 2004.
9. L. Massouli, A. Twigg, C. Gkantsidis, and P. Rodriguez. "Randomized decentralized broadcasting algorithms" *INFOCOM, 2007*, Anchorage, AK, May 2007.
10. S. Sanghavi, B. Hajek, and L. Massouli. "Gossiping with multiple messages" *INFOCOM, 2007* Anchorage, AK, May 2007.
11. T. Bonald, L. Massouli, F. Mathieu, D. Perino, A. Twigg, "Epidemic Live Streaming: Optimal Performance Trade-Offs", *Sigmetrics 2008*, Annapolis, ML, June 2008.
12. A. P. C.da Silva, E. Leonardi, M. Mellia, M. Meo. "A Bandwidth-Aware Scheduling Strategy for P2P-TV Systems", *8th International Conference on Peer-to-Peer Computing 2008 (P2P'08)*, Aachen, September 8-11, 2008.
13. Yong Liu,"On the minimum delay peer-to-peer video streaming: how realtime can it be?", *ACM Multimedia 2007*, Augsburg, Germany, September 2007.
14. P2PTV-Sim, <http://www.napa-wine.eu/cgi-bin/twiki/view/Public/Software>.